

## IMPLEMENTASI SISTEM CLIENT-SERVER BERBASIS FLASK DAN MYSQL UNTUK PENGELOLAAN DATA MAHASISWA DENGAN PROTOKOL TCP/IP

Ratnasari, Nur Hafiizah, Ni Wayan Erdiani, Asa Hari Wibowo

Program Studi Teknik Informatika S1, Fakultas Teknik, Universitas Halu Oleo,  
Kampus Hijau Bumi Tridharma, Anduonohu, Kec. Kambu, Kota Kendari, Indonesia  
ratnasari200422@gmail.com

### ABSTRAK

Sistem *client-server* berbasis web berbasis server menjadi salah satu solusi efektif untuk pengelolaan data yang terdistribusi. Pada penelitian ini dibuat sebuah sistem pengelolaan data mahasiswa menggunakan framework flask dengan menggunakan framework Flask sebagai web server, MySQL sebagai database, dan komunikasi berbasis protokol TCP/IP melalui HTTP. Permasalahan yang dihadapi adalah bagaimana merancang dan mengimplementasikan sistem yang efisien untuk menyimpan dan mengakses data siswa baik dari sisi klien maupun sisi server. Tujuan dari penelitian ini adalah menghasilkan suatu sistem pengolahan data yang dapat mengelola data yang dapat memproses input dari form web, menyimpan data tersebut di MySQL dan menampilkan data dengan cepat dan aman. Metode yang digunakan adalah metode waterfall yang meliputi tahapan analisis, perancangan, implementasi, pengujian dan pemeliharaan sistem. Hasil pengujian menunjukkan bahwa sistem dapat menangani data mahasiswa dengan baik, dan komunikasi antara klien dan server berjalan efektif melalui protokol TCP/IP. Penelitian ini memberikan kontribusi untuk mengimplementasikan sistem pengolahan data berbasis web dengan arsitektur *client-server* yang dapat digunakan pada aplikasi lain.

**Kata kunci :** *client-server, data mahasiswa, tcp/ip, web.*

### 1. PENDAHULUAN

Di era digital saat ini, pengelolaan data yang baik menjadi salah satu kebutuhan terpenting di berbagai sektor, termasuk lingkungan akademis. Sistem pengolahan data mahasiswa merupakan salah satu contoh aplikasi yang memerlukan perangkat distribusi dan akses yang cepat, aman, dan efisien. Salah satu pendekatan yang paling banyak digunakan dalam pengembangan sistem pengolahan data adalah arsitektur *client-server*, yang menggunakan web sebagai antarmuka dan database sebagai pusat penyimpanannya. Dalam konteks ini, framework Flask digunakan sebagai server web karena kemudahannya pengembangannya dan kemampuannya menangani permintaan dan respons HTTP. Di sisi lain, MySQL digunakan sebagai sistem manajemen basis data yang dapat menyimpan data dalam jumlah besar secara terstruktur. Komunikasi antara klien dan server terjadi melalui protokol TCP/IP, yang memastikan bahwa data dapat dikirim dan diterima dengan andal. Permasalahan utama dalam penelitian ini adalah bagaimana membuat suatu sistem yang dapat mengelola data siswa secara efisien mulai dari pengisian data mahasiswa hingga penyimpanan dan pengambilan data serta server pada database MySQL. Selain itu, penelitian ini juga bertujuan untuk memastikan bahwa sistem yang dibangun dapat berkomunikasi melalui protokol TCP/IP, sehingga memudahkan proses pemasukan dan pengelolaan data yang terdistribusi. Oleh karena itu, penelitian ini bertujuan untuk merancang dan mengimplementasikan sistem pengolahan data mahasiswa berbasis *client-server* menggunakan protokol Flask, MySQL dan TCP/IP.

Salah satu kekurangan utama dari sistem ini adalah skala terbatas. Sistem berbasis Flask dan MySQL ini cocok untuk lingkungan kecil, tetapi saat jumlah pengguna atau volume data meningkat, performa sistem dapat menurun karena Flask tidak dirancang untuk menangani beban besar tanpa konfigurasi atau migrasi ke platform yang lebih kuat. Berdasarkan latar belakang tersebut, penulis mengangkat kasus ini menjadi skripsi dengan mengambil judul: "Implementasi Sistem *Client-Server* Berbasis Flask dan MySQL untuk Pengelolaan Data Mahasiswa dengan Protokol TCP/IP" yang bertujuan untuk merancang dan mengembangkan sistem pengelolaan data mahasiswa yang efisien, aman, dan terintegrasi[1]. Sistem ini diharapkan mampu memfasilitasi proses input, penyimpanan, dan pengambilan data mahasiswa melalui protokol TCP/IP dengan menggunakan Flask sebagai server web dan MySQL sebagai basis data. Selain itu, penelitian ini bertujuan untuk memastikan bahwa komunikasi antara *client* dan server berjalan dengan baik serta mendukung kinerja sistem yang stabil.

### 2. TINJAUAN PUSTAKA

Penelitian sebelumnya telah membuktikan bahwa penerapan protokol TCP/IP dalam komunikasi antara klien dan server dapat meningkatkan efisiensi serta keandalan sistem. Hal ini terutama terlihat dalam aplikasi yang terkait dengan basis data. Studi oleh Yones Raven dan Teady Matius Surya Mulyana menunjukkan bahwa protokol TCP/IP memungkinkan Database Access beroperasi sebagai server database. Permintaan SQL yang dikirim dari *client* akan diterima dan diproses oleh server, lalu hasilnya akan dikirim kembali ke *client*. Dengan memanfaatkan socket,

aplikasi server bisa tetap dalam keadaan pasif hingga menerima permintaan dari klien, yang selanjutnya diproses dan dibalas secara efisien. Hal ini menjadikan TCP/IP pilihan ideal dalam sistem yang membutuhkan stabilitas komunikasi jaringan.

Penelitian terdahulu telah menunjukkan bahwa TCP/IP dengan socket programming memberikan efisiensi tinggi dalam komunikasi antara klien dan server, terutama dalam hal pengiriman dan pengolahan data pada basis data. Penelitian yang dilakukan oleh Iedam Fardian Anshori pada tahun 2019 mengungkapkan bahwa socket memfasilitasi transfer data antara aplikasi client dan server secara langsung melalui kombinasi host-address dan port address. Hal ini terbukti efektif terutama dalam aplikasi pengiriman file teks ke dalam basis data. Dengan TCP/IP, server dapat "mendengarkan" koneksi yang datang dari client di port yang ditentukan dan memproses permintaan data tersebut. Hasil pengujian telah menyimpulkan bahwa TCP/IP layak digunakan sebagai sarana pengiriman data karena mampu menjamin koneksi yang dapat diandalkan dan alur data yang teratur di antara kedua aplikasi. Sementara itu, TCP/IP sebagai protokol yang berorientasi pada koneksi, sebagaimana telah disebutkan oleh Tanenbaum dan Listiyono (2009), memiliki tujuh fitur utama. Fitur tersebut meliputi koneksi full-duplex dan keandalan pengiriman data tanpa duplikasi atau kehilangan. Fitur ini memungkinkan aplikasi server untuk menangani transmisi data secara efisien, bahkan dalam jaringan yang cenderung tidak stabil. Selain itu, penelitian sebelumnya mengonfirmasi bahwa implementasi TCP/IP dengan Java socket, seperti yang dikemukakan oleh Sugiarto memungkinkan aplikasi untuk diadaptasi ke berbagai sistem karena sifat platform-independen Java. Integrasi TCP/IP dalam konteks aplikasi big data seperti analisis sentimen berita pun dapat berjalan efektif berkat kestabilan transmisi data yang ditawarkan protokol ini. Hal tersebut mendukung pemrosesan data dalam jumlah besar secara akurat dan cepat[3].

### 2.1. Client-server

Terdapat beberapa definisi Client-Server Menurut beberapa ahli antara lain, Client-Server adalah client mengirimkan permintaan ke server, server menerjemahkan pesan, kemudian berusaha memenuhi permintaan. Client-Server adalah suatu arsitektur dimana sumberdaya server menyediakan komputasi untuk banyak komponen client. Client dan server bisa berjalan pada mesin yang sama atau berbeda, ditulis dalam berbagai bahasa dan menggunakan sistem operasi yang berbeda[4]. Adapun jenis arsitektur Client-Server dikategorikan menjadi dua bagian arsitektur diantara lainnya yaitu:

### 2.2. Arsitektur 2-tier

Arsitektur Client-Server 2-tier adalah lingkungan Client-Server tradisional. Client biasanya terletak pada workstation yang digunakan langsung oleh user dan

dibuat dengan bahasa pemrograman seperti VB, Delphi, Powerbuilder, Foxpro dan lain-lain[5]. Server adalah sebuah komputer yang diletakkan dibagian lain pada jaringan yang menjalankan software database sever seperti oracle, SQL Server, MySQL dan sebagainya.

### 2.3. Arsitektur 3-tier

Arsitektur Client-Server 3-tier memecah komponen kedalam tiga kategori atau lapisan. Pada arsitektur 3-tier terdapat lapisan tambahan middletier yang bertugas mengelola application/business logic, misalnya: rumus perhitungan gaji, pajak, bunga, deposito dan lain lain. Software middletier tersebut sebagai contoh: MTS (Microsoft transaction server), EJB dan lain lain.

### 2.4. Tcp/ip

Transmission Control Protocol (TCP) adalah salah satu jenis protocol yang memungkinkan kumpulan komputer untuk berkomunikasi dan bertukar data dalam suatu network atau jaringan. TCP merupakan suatu protokol yang berada di lapisan transport (baik itu dalam tujuh lapis model referensi OSI atau model DARPA) yang berorientasi sambungan (Connection Oriented) dan dapat diandalkan (reliable). Kerugian dari TCP adalah proses transfer data menjadi agak lambat karena diperlukan proses *Three ways Handshake*[6], adalah standar utama yang digunakan untuk komunikasi data di internet. TCP/IP terdiri dari beberapa lapisan yang bertanggung jawab untuk memastikan pengiriman data yang andal dari satu perangkat ke perangkat lainnya dalam jaringan. Protokol ini sangat penting dalam sistem client-server karena menjamin komunikasi yang stabil dan aman antara client dan server. Pada sistem ini, protokol TCP/IP digunakan untuk mengirimkan data dari client (melalui HTTP) ke server Flask, yang kemudian mengolahnya dan berinteraksi dengan database MySQL[3].

### 2.5. Mysql

MySQL adalah salah satu jenis database server yang sangat populer, hal ini disebabkan karena MySQL menggunakan SQL sebagai bahasa dasar untuk mengakses databasenya. MySQL bersifat Open Source, Software ini dilengkapi dengan Source code (kode yang dipakai untuk membuat MySQL). sebagai salah satu sistem manajemen basis data relasional, banyak digunakan dalam berbagai aplikasi berbasis web karena keandalan dan skalabilitasnya. MySQL memungkinkan pengelolaan data dalam jumlah besar secara efisien, dengan dukungan berbagai fitur keamanan dan pemeliharaan data. Dalam konteks sistem pengelolaan data mahasiswa, MySQL berperan sebagai penyimpanan terpusat yang memungkinkan data diakses dengan cepat oleh server dan dimanipulasi melalui query SQL.

### 3. METODE PENELITIAN

Pada Penelitian ini menggunakan metode pengembangan sistem dengan pendekatan Waterfall, yang terdiri dari beberapa tahapan sebagai berikut:

#### 3.1. Tahap Analisis Kebutuhan

Pada tahap ini, dilakukan identifikasi kebutuhan sistem pengelolaan data mahasiswa. Kebutuhan yang diidentifikasi meliputi kebutuhan akan user interface (form input mahasiswa), koneksi ke database MySQL untuk penyimpanan dan pengambilan data, serta protokol komunikasi yang digunakan, yaitu TCP/IP melalui HTTP[7].

#### 3.2. Tahap Perancangan Sistem

Perancangan sistem dilakukan dengan merancang arsitektur client-server. Pada sisi client, dibuat form HTML sebagai antarmuka pengguna yang dihubungkan dengan server Flask melalui metode POST[8]. Pada sisi server, Flask dirancang untuk menangani request dan respons dari client, serta mengelola koneksi ke database MySQL. Database MySQL dirancang untuk menyimpan data mahasiswa dengan kolom-kolom seperti nama, NIM, dan jurusan. Protokol TCP/IP digunakan untuk menjamin komunikasi antara client dan server.

#### 3.3. Tahap Implementasi

Implementasi dilakukan dengan mengembangkan aplikasi web berbasis Flask untuk server dan MySQL sebagai basis data. Komunikasi antara client dan server diimplementasikan menggunakan protokol TCP/IP melalui HTTP. Flask berfungsi sebagai server web yang mengelola permintaan dari client (input data mahasiswa), memproses query SQL, dan mengembalikan hasil query dari database MySQL[9].

#### 3.4. Tahap Pengujian Sistem

Pengujian dilakukan untuk memastikan bahwa sistem dapat berjalan dengan baik sesuai fungsinya. Pengujian fungsional dilakukan dengan memeriksa apakah data yang diinput melalui form web dapat disimpan dan diambil dari MySQL dengan benar[10]. Pengujian juga dilakukan untuk memastikan komunikasi antar client dan server melalui protokol TCP/IP berjalan secara efektif tanpa adanya gangguan atau kehilangan data.

#### 3.5. Tahap Pemeliharaan

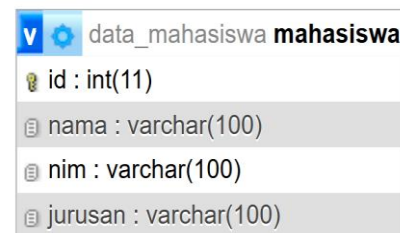
Setelah sistem diimplementasikan, tahap pemeliharaan dilakukan untuk memantau performa sistem dan memperbaiki bug yang mungkin muncul. Sistem juga diuji kembali secara berkala untuk memastikan bahwa perubahan atau peningkatan yang dilakukan tidak mempengaruhi kinerja keseluruhan[11].

### 4. HASIL DAN PEMBAHASAN

Pada implementasi sistem client-server berbasis TCP/IP yang telah dikembangkan, terdapat beberapa komponen utama yang berfungsi untuk mendukung pengelolaan data mahasiswa. Berikut adalah pembahasan mengenai tampilan aplikasi client dan server, serta pengujian sistem.

#### 4.1. Tampilan Desain Database

Tabel mahasiswa dalam basis data data\_mahasiswa berfungsi sebagai penyimpanan utama untuk informasi terkait mahasiswa. Desain tabel ini sangat penting dalam sistem manajemen akademik, terutama untuk aplikasi kampus atau universitas yang membutuhkan pengelolaan data mahasiswa secara terstruktur dan mudah diakses.



| data_mahasiswa         |
|------------------------|
| id : int(11)           |
| nama : varchar(100)    |
| nim : varchar(100)     |
| jurusan : varchar(100) |

Gambar 1. Tampilan desain database

Tabel ini memiliki empat kolom utama, yaitu:

- id: Kolom id menggunakan tipe data int dengan panjang 11 karakter. Kolom ini berfungsi sebagai primary key (kunci utama) dari tabel mahasiswa, yang berarti setiap nilai dalam kolom ini harus unik dan tidak boleh kosong. Primary key berguna untuk memastikan bahwa setiap entri dalam tabel ini dapat diidentifikasi secara unik. Dengan adanya kolom id, sistem dapat menghindari duplikasi data dan memungkinkan setiap data mahasiswa untuk memiliki identifikasi yang unik.
- nama: Kolom nama memiliki tipe data varchar dengan panjang maksimal 100 karakter. Kolom ini menyimpan nama mahasiswa dalam bentuk teks dan bertujuan untuk mencatat nama lengkap setiap mahasiswa. Penggunaan tipe data varchar memberikan fleksibilitas dalam panjang teks, sehingga dapat menyimpan nama mahasiswa dengan berbagai jumlah karakter.
- nim: Kolom nim (Nomor Induk Mahasiswa) juga menggunakan tipe data varchar dengan panjang maksimal 100 karakter. Kolom ini berfungsi untuk menyimpan NIM yang bersifat unik untuk setiap mahasiswa. NIM biasanya digunakan sebagai identitas resmi mahasiswa yang diberikan oleh institusi pendidikan, yang membedakan satu mahasiswa dengan mahasiswa lainnya. NIM sangat penting dalam sistem akademik karena sering digunakan dalam proses administrasi, seperti pendaftaran mata kuliah, pencatatan nilai, dan keperluan akademik lainnya.
- jurusan: Kolom jurusan juga memiliki tipe data varchar dengan panjang maksimal 100 karakter dan digunakan untuk menyimpan informasi

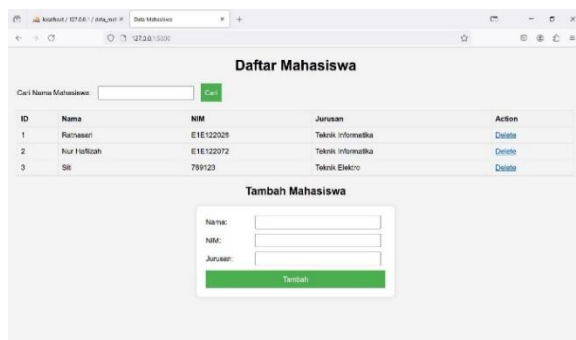
mengenai jurusan atau program studi yang diambil oleh mahasiswa. Informasi ini berguna untuk mengelompokkan mahasiswa berdasarkan bidang studi atau program yang mereka pilih, sehingga memungkinkan pengelolaan data berdasarkan jurusan.

Secara keseluruhan, desain tabel mahasiswa ini membantu sistem basis data untuk menyimpan data mahasiswa dengan terstruktur, memastikan keunikan data dengan adanya primary key id, dan memudahkan proses pengelolaan informasi terkait mahasiswa dalam konteks akademik. Desain ini sederhana namun efektif untuk memenuhi kebutuhan dasar penyimpanan data mahasiswa di lingkungan perguruan tinggi.

#### 4.2. Tampilan Aplikasi Client

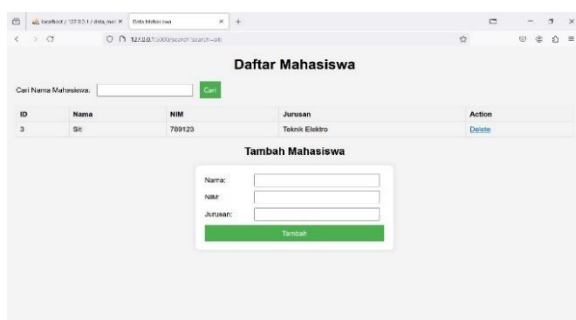
Pada aplikasi client, terdapat form input data memasukkan informasi mahasiswa, seperti nama, NIM, dan jurusan, serta tombol submit untuk mengirimkan data ke server. Tampilan ini dirancang menggunakan form HTML yang terhubung dengan server Flask melalui protokol TCP/IP. Pada saat tombol cari ditekan, data yang dimasukkan oleh pengguna akan dikirimkan ke server untuk disimpan di basis data MySQL.

Gambar berikut menampilkan antarmuka aplikasi client:



Gambar 2. Tampilan aplikasi client

Selain itu, aplikasi client juga menyediakan fitur untuk melakukan pencarian data mahasiswa yang sudah tersimpan. Pencarian dapat dilakukan berdasarkan nama atau NIM mahasiswa, dan hasilnya dapat dilihat pada gambar berikut:



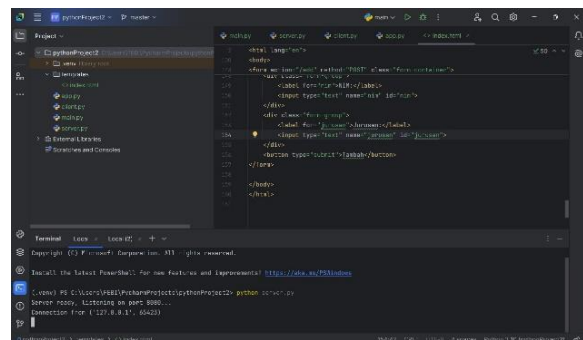
Gambar 3. Tampilan fitur pencarian data mahasiswa

Tombol delete disediakan untuk menghapus semua input serta membersihkan tampilan tabel hasil pencarian.

#### 4.3. Tampilan server

Pada sisi server, aplikasi yang dikembangkan menggunakan Flask untuk menerima dan merespons permintaan dari client. Server berfungsi untuk menjalankan query SQL berdasarkan permintaan client dan menampilkan hasilnya. Server juga menampilkan informasi tentang alamat client yang terkoneksi serta perintah SQL yang dijalankan.

Pada sisi server, aplikasi menggunakan Flask untuk menangani komunikasi antara client dan server dengan menggunakan protokol TCP/IP. Server memiliki beberapa fungsi utama dalam sistem ini, yaitu menerima permintaan dari client, menjalankan query SQL ke MySQL, dan mengembalikan hasilnya ke client.

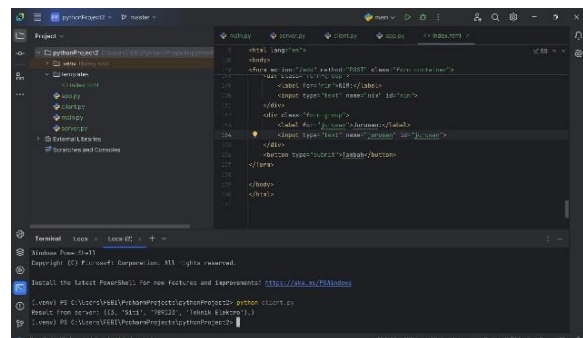


Gambar 4. Tampilan server

Gambar ini menunjukkan antarmuka dasar dari aplikasi server yang dijalankan di PyCharm, memperlihatkan informasi tentang koneksi klien yang aktif serta status server yang sedang mendengarkan pada port 8080. Aplikasi server ini mendukung koneksi berbasis TCP/IP untuk menerima data dari klien.

#### 4.4. Tampilan Client

Server memiliki beberapa fungsi utama dalam sistem ini, yaitu menerima permintaan dari client, menjalankan query SQL ke MySQL, dan mengembalikan hasilnya ke client. Berikut adalah tampilan dari sisi clientnya.

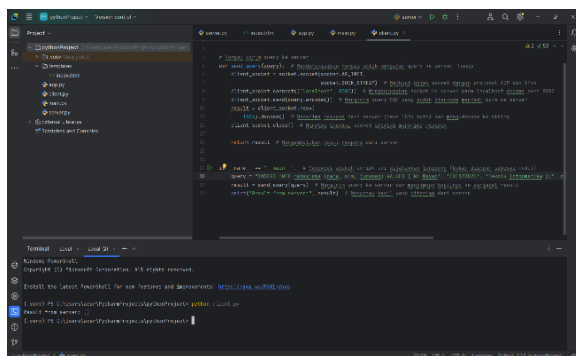


Gambar 5. Tampilan pada aplikasi client

Gambar ini menunjukkan antarmuka PyCharm dengan proyek Python yang sedang berjalan. Di bagian terminal, terlihat bahwa skrip `client.py` telah dieksekusi, dan server mengembalikan hasil berupa data mahasiswa dengan format tuple: (3, 'Siti', '789123', 'Teknik Elektro'). Ini menunjukkan bahwa komunikasi antara klien dan server berhasil dilakukan melalui koneksi berbasis TCP/IP.

#### 4.5. Pengujian Sistem

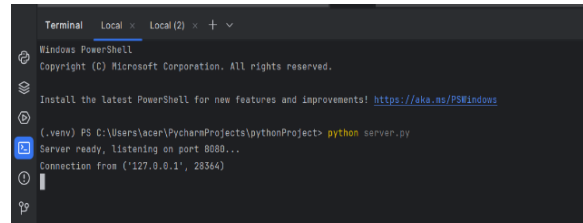
Pengujian aplikasi client-server berbasis Flask dan MySQL untuk pengelolaan data mahasiswa dilakukan untuk memastikan fungsi dan kinerjanya. Pengujian meliputi black box testing pada fitur CRUD (create, read, update, delete) untuk memastikan setiap fungsi berjalan sesuai spesifikasi, serta uji kinerja untuk mengukur waktu respons server terhadap permintaan klien melalui protokol TCP/IP. Rata-rata waktu respons berada di rentang optimal, menunjukkan bahwa sistem mampu mengelola data mahasiswa dengan cepat dan efisien. Selain itu, uji beban dilakukan untuk menilai keandalan sistem dalam menangani permintaan simultan dari beberapa klien, yang menunjukkan bahwa aplikasi tetap stabil tanpa penurunan performa saat beban meningkat.



Gambar 6. Tampilan pada client

Pada gambar ini menunjukkan bahwa client dan server berkomunikasi untuk menambahkan data ke database. Di file `client.py`, fungsi `send_query()` dibuat untuk mengirimkan query SQL ke server melalui koneksi TCP. Setelah client menjalankan `client.py`, ia akan membuat query SQL `INSERT INTO mahasiswa (nama, nim, jurusan) VALUES ('Ni Wayan', 'E1E122024', 'Teknik Informatika');` yang bertujuan menambahkan data baru ke dalam tabel mahasiswa di database. Query ini kemudian dikirim ke server.

Server akan menerima query tersebut dan menjalankannya untuk menyimpan data ke dalam database. Setelah query dieksekusi, server mengirimkan respon kembali ke client, yang dalam hal ini berupa hasil kosong (`[]`). Hasil ini dicetak oleh client sebagai "Result from server: []" yang menunjukkan bahwa data sudah dikirim, meskipun responnya tidak menampilkan hasil. Alur ini secara keseluruhan menunjukkan proses penambahan data dari client ke database melalui server.



Gambar 7. Tampilan pada server

#### 5. KESIMPULAN DAN SARAN

Kesimpulan yang diperoleh dari penelitian ini adalah bahwa sistem pengelolaan data mahasiswa berbasis Flask, MySQL, dan TCP/IP berhasil diimplementasikan dengan baik dan mampu memproses input data dari client serta menyimpan dan mengelola data mahasiswa di basis data MySQL secara efektif. Sistem ini cocok untuk skala kecil hingga menengah, namun memiliki keterbatasan dalam hal skalabilitas dan keamanan. Untuk penelitian selanjutnya, disarankan untuk mengoptimalkan sistem dengan menggunakan framework yang lebih skalabel dan menerapkan enkripsi komunikasi data menggunakan protokol SSL/TLS untuk meningkatkan keamanan.

#### DAFTAR PUSTAKA

- [1] S. Fadli and K. Imtihan, "Analisis Dan Perancangan Sistem Administrasi Dan Transaksi Berbasis Client Server," *J. Inform. dan Rekayasa Elektron.*, vol. 1, no. 2, p. 7, 2018, doi: 10.36595/jire.v1i2.54.
- [2] Y. Raven and T. M. S. Mulyana, "Implementasi Tcp/Ip Untuk Membuat Server Database Access," *J. Teknol. Inf.*, vol. 11, pp. 44–50, 2017, [Online]. Available: <https://journal.ubm.ac.id/index.php/teknologi-informasi/article/view/260>
- [3] I. F. Anshori, "Implementasi Socket Tcp/Ip Untuk Mengirim Dan Memasukan File Text Kedalam Database," *Responsif*, vol. Vol 1 No 1, no. 1, pp. 1–5, 2019.
- [4] R. Affairs, "Implementasi Protokol Komunikasi Modbus Untuk Mini Scada Pada Plat Pengisian Serbuk Temulawak," vol. 2, no. 5, pp. 314–326, 2024.
- [5] A. Bahardiansyah, R. Yulianto, and N. Novitasari, "Analisis Dan Perancangan Sistem Informasi Pada Coffee Shop Songot Berbasis Client Server," p. 21, 2019.
- [6] C. P. Bandhaso and Y. Sutanto, "Analisis Dan Implementasi Metode Random Early Detection (Red) Pada Jaringan TCP/IP," *Respati*, vol. 17, no. 2, p. 51, 2022, doi: 10.35842/jtir.v17i2.458.
- [7] T. Natanegara, Y. Muhyidin, and D. Singasatia, "Implementasi Honeypot Cowrie Dan Snort Sebagai Alat Deteksi Serangan Pada Server," *JATI (Jurnal Mhs. Tek. Inform.)*, vol. 7, no. 3, pp. 1871–1877, 2023, doi: 10.36040/jati.v7i3.6989.
- [8] 2021 : 41) Berliana, "Analisis Perbandingan Kinerja Aplikasi Web Browser Berdasarkan

- Sistem Operasi,” *Dasar-Dasar Ilmu Polit.*, no. 1, pp. 1–15, 2021.
- [9] G. E. Yogiswara, “Aplikasi Sistem Pakar Diagnosa Kegagalan Koneksi TCP/IP Menggunakan Metode Forward Chaining,” *Pros. Semin. Nas. Teknol. ...*, pp. 453–462, 2022, [Online]. Available: <http://prosiding.unipma.ac.id/index.php/SENATIK/article/view/2899%0Ahttp://prosiding.unipma.ac.id/index.php/SENATIK/article/download/2899/2921>
- [10] Rudy Adipranata, “Implementasi Protokol Tcp/Ip Untuk Pengendalian Komputer Jarak Jauh,” *J. Inform.*, vol. 3, no. 1, pp. 30–36, 2002, [Online]. Available: <http://puslit2.petra.ac.id/ejournal/index.php/inf/article/view/15814>
- [11] B. Y. Pratama and E. Anisa, “Analisis Implementasi Komunikasi Modbus TCP/IP dalam Penerapan Visualisasi Data Hasil Produksi pada Sistem Andon Line Production,” *J. Asimetrik J. Ilm. Rekayasa Inov.*, vol. 5, pp. 43–52, 2023, doi: 10.35814/asiimetrik.v5i1.3791.